1. **Поясните суть паттерна *Singleton,* что дает его** **применение.**

Паттерн Singleton представляет собой порождающий паттерн проектирования, который гарантирует, что в системе будет существовать только один экземпляр определенного класса, и предоставляет глобальную точку доступа к этому экземпляру.

1. **Поясните реализацию класса *Logger:***

* статический метод create (почему статический);

Метод Create() является статическим, чтобы обеспечить доступ к единственному экземпляру класса Logger без необходимости создавать новые объекты. Это позволяет использовать класс Logger из разных частей программы, вызывая метод Create() для получения экземпляра.

* приватный конструктор (почему приватный);

Конструктор класса Logger объявлен как приватный (private), что означает, что он не может быть вызван извне класса. Такое ограничение гарантирует, что объекты класса Logger могут быть созданы только внутри самого класса, и только один раз. Это важно для обеспечения наличия только одного экземпляра класса Logger в программе, что соответствует паттерну Singleton.

* поясните жизненный цикл ***Singleton****-*класса на тестовом примере.

При вызове статического метода Create(), он проверяет, существует ли уже экземпляр класса Logger. Если экземпляр уже создан, метод просто возвращает его. Если экземпляр не существует, создается новый экземпляр класса Logger и сохраняется в статической переменной \_logger. Последующие вызовы метода Create() будут возвращать тот же самый экземпляр класса Logger.

1. **Приведите примеры приложений, в которых может быть** **применен паттерн *Singleton.***

* Логгирование: Singleton Logger обеспечивает централизованную запись логов в приложении, позволяя различным компонентам логировать события и ошибки в один и тот же файл или другой источник логов.
* Кэширование: Singleton Cache может использоваться для хранения временных данных или результата выполнения операций, чтобы избежать повторных вычислений или запросов к внешним источникам. Он предоставляет единственный объект кэша, доступный из разных частей приложения.
* Пул подключений к БД: Singleton Connection Pool управляет пулом активных подключений к базе данных. Он гарантирует, что доступ к подключениям осуществляется через единственный объект, контролирующий создание, выдачу и возврат подключений.
* Настройка приложения: Singleton Configuration используется для хранения и доступа к глобальным настройкам приложения, таким как параметры подключения к базе данных, пути к файлам конфигурации и другие конфигурационные параметры.
* Управление ресурсами: Singleton ResourceManager централизованно управляет доступом к разделяемым ресурсам в приложении, таким как печать, работа с файлами или сетевые соединения.